PEG Parsing in Python 3.9

Tomáš Dacík
xdacik00@stud.fit.vutbr.cz

Ondřej Kinšt
xkinst01@stud.fit.vutbr.cz

Before its version 3.9, CPython (most popular implementation of the Python programming language) used a LL(1)-based parser. Due to limitations of LL grammars, some syntactic features required additional and sometimes very complicated workarounds implemented outside the grammar. Moreover, the original LL grammar also contained a lot of unnatural rules obtained e.g. by removal of the left recursion.

To improve these problems, the parser was replaced by a PEG-based (*Parsing Expression Grammar*) parser. The main idea behind PEG is that it allows an unbounded lookahead that is achieved by an ordering of right-hand sides of grammar rules and use of a backtracking when more than a single rule can be selected. Such parser either reports an error or finds the first valid parse tree. This means that PEG cannot be ambiguous and some syntactic constructs can be therefore described more easily and in a more readable way.

We will further discuss an efficient implementation method called *packrat parsing* that uses a memoization to improve the exponential time complexity of a naive parsing algorithm to a linear running time.